WE CLAIM:

1. A computer-implemented method for detecting errors in a computer system comprising a plurality of clusters, each cluster including a plurality of local nodes and an interconnection controller interconnected by point-to-point intra-cluster links, communications between the local nodes and the interconnection controller made via an intra-cluster protocol using intra-cluster packets, the interconnection controller of each cluster interconnected by point-to-point inter-cluster links with the interconnection controller of other clusters, the computer-implemented method comprising:

forming an inter-cluster packet by encapsulating an intra-cluster packet;

encoding a sequence identifier in the inter-cluster packet;

calculating first cyclic redundancy code check data based only upon the inter-cluster packet;

encoding the first cyclic redundancy code check data in the inter-cluster packet; and

transmitting the inter-cluster packet from a first cluster to a second cluster on a point-to-point inter-cluster link.

2. The computer-implemented method of claim 1, wherein the encoding steps comprise encoding in an area of the inter-cluster packet reserved for link layer information.

3. The computer-implemented method of claim 1, further comprising:

receiving the inter-cluster packet; and

calculating second cyclic redundancy code check data based only upon the inter-cluster packet.

4. The computer-implemented method of claim 3, further comprising:

detecting an error in the inter-cluster packet based upon the second cyclic redundancy code check data; and

notifying the first cluster of the error.

5. An apparatus for detecting errors in a computer system comprising a plurality of clusters, each cluster including a plurality of local nodes and an interconnection controller interconnected by point-to-point intra-cluster links, communications between the local nodes and the interconnection controller made via an intra-cluster protocol using intra-cluster packets, the interconnection controller of each cluster interconnected by point-to-point inter-cluster links with the interconnection controller of other clusters, the apparatus comprising:

means for forming an inter-cluster packet by encapsulating an intra-cluster packet;

means for encoding a sequence identifier in the inter-cluster packet;

means for calculating first cyclic redundancy code check data based only upon the inter-cluster packet;

means for encoding the first cyclic redundancy code check data in the inter-cluster packet; and

means for transmitting the inter-cluster packet from a first interconnection controller to a second interconnection controller on a point-to-point inter-cluster link.

6. A computer system, comprising:

a first cluster including a first plurality of processors and a first interconnection controller, the first plurality of processors and the first interconnection controller interconnected by first point-to-point intra-cluster links; and

a second cluster including a second plurality of processors and a second interconnection controller, the second plurality of processors and the second interconnection controller interconnected by second point-to-point intra-cluster links, the first

interconnection controller coupled to the second interconnection controller by point-to-point inter-cluster links, communications on the first and second intra-cluster links made via an intra-cluster protocol by intra-cluster packets;

wherein the first interconnection controller is configured to:

receive an intra-cluster packet from a first processor in the first plurality of processors;

store the intra-cluster packet in a buffer;

add a header, including a sequence identifier, to the intra-cluster packet to form a high-speed link packet;

compute a first cyclic redundancy code check based only upon the high-speed link packet;

encode first cyclic redundancy code check data in the high-speed link packet; and

transmit the high-speed link packet to the second interconnection controller in the second cluster;

wherein the second interconnection controller is configured to:

receive the high-speed link packet;

compute a second cyclic redundancy code check based only upon the high-speed link packet;

compare results of the second cyclic redundancy code check with the encoded first cyclic redundancy code check data in the high-speed link packet; and

notify the first interconnection controller regarding the results of the comparison.

7. The computer system of claim 6, wherein the second interconnection controller is further configured to detect a gap in sequence identifiers of high-speed link packets received from the first interconnection controller, and wherein the notifying step comprises notifying the first interconnection controller of the gap.

8. The computer system of claim 6, wherein the notifying step comprises sending an ACK packet to the first interconnection controller indicating a sequence identifier of one or more high-speed link packets received without errors.

9. The computer system of claim 6, wherein the notifying step comprises sending a NACK packet to the first interconnection controller indicating a sequence identifier of one or more high-speed link packets received with errors.

10. The computer system of claim 6, wherein the notifying step comprises initiating a retry sequence when an error is detected in the comparing step, the retry sequence causing the first interconnection controller to transmit high-speed link packets that include copies of intra-cluster packets stored in the buffer.

11. The computer system of claim 6, wherein the notifying step comprises initiating a retry sequence when a gap is detected, the retry sequence causing the first interconnection controller to transmit high-speed link packets that include copies of intra-cluster packets stored in the buffer.

12. The computer system of claim 8, wherein the first interconnection controller is further configured to purge intra-cluster packets stored in the buffer that correspond to the sequence identifiers of the ACK packets.

13. The computer system of claim 10, further comprising the step of sending, after the retry sequence is complete, an acknowledgement packet with a sequence identifier of a last packet received without error.

14. The computer system of claim 10, further comprising the step of sending, after the retry sequence is complete, an negative acknowledgement packet with a sequence identifier of a last packet received having an error.

15. The computer system of claim 10, wherein the retry sequence provides a one-time, error-free delivery of protocol layer packets.

16. An interconnection controller, comprising:

an intra-cluster interface configured for coupling with intra-cluster links to a plurality of local processors arranged in a point-to-point architecture in a local cluster;

an inter-cluster interface configured for coupling with an inter-cluster link to a non-local interconnection controller in a non-local cluster;

a transceiver configured to:

receive an intra-cluster packet from a local processor via an intra-cluster link;

encode a sequence identifier in a header of the intra-cluster packet;

compute cyclic redundancy code check data based only on the encoded packet; and

encode the cyclic redundancy code check data in the encoded packet; and

a serializer/deserializer configured to serialize the encoded packet and forward the encoded, serialized packet to the inter-cluster interface for transmission to the non-local interconnection controller via an inter-cluster link.

17. The interconnection controller of claim 16, wherein the inter-cluster interface is further configured to receive encoded, serialized packets from the non-local interconnection controller, wherein the serializer/deserializer is further configured to deserialize the encoded, serialized packets and wherein the transceiver is further configured to perform a cyclic redundancy code check on the deserialized packets.

18. An integrated circuit comprising the interconnection controller of claim 16.

19. A set of semiconductor processing masks representative of at least a portion of the interconnection controller of claim 16.

20. At least one computer-readable medium having data structures stored therein representative of the interconnection controller of claim 16.

21. The integrated circuit of claim 18, wherein the integrated circuit comprises an application-specific integrated circuit.

22. The at least one computer-readable medium of claim 20, wherein the data structures comprise a simulatable representation of the interconnection controller.

23. The at least one computer-readable medium of claim 20, wherein the data structures comprise a code description of the interconnection controller.

24. The at least one computer-readable medium of claim 22, wherein the simulatable representation comprises a netlist.

25. The at least one computer-readable medium of claim 23, wherein the code description corresponds to a hardware description language.

5